



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

IS PRO SPRÁVU BYTOVÉHO DOMU

IS FOR ADMINISTRATION OF APARTMENT BLOCK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR FOTIJEV

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN LUHAN, Ph.D.

BRNO 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Fotijev Petr

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

IS pro správu bytového domu

v anglickém jazyce:

IS for Administration of Apartment Block

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení a jejich přínos

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

BÉBR, R. a P. DOUCEK. Informační systémy pro podporu manažerské práce. 1. vyd. Praha: Profesional Publishing, 2005. 223 s. ISBN 80-86419-79-7.

BRÖHMER, M. Návrhové vzory v php. 1. vyd. Brno: Computer Press, 2012. 320 s. ISBN 978-80-251-3338-5.

DOUCEK, P. Řízení projektů informačních systémů. 1. vyd. Praha: Profesional Publishing, 2004. 163 s. ISBN 80-86419-71-1.

MOLNÁR, Z. Efektivnost informačních systémů. 1. vyd. Praha: Grada, 2000. 142 s. ISBN 80-7169-410-X.

PALETA, P. Co programátory ve škole neučí. 1. vyd. Brno: Computer Press, 2003. 337 s. ISBN 80-251-0073-1.

Vedoucí bakalářské práce: Ing. Jan Luhan, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 31.05.2013

Abstrakt

Bakalářská práce se zabývá analýzou požadavků na vytvoření informačního systému pro správu bytového domu, návrhem řešení vycházejícího z vytvořené analýzy a implementace části tohoto návrhu.

Cílem práce je vytvořit funkční návrh informační systém, který uspokojí požadavky majitele domu.

Abstract

This bachelor thesis deals with analysis of requirements for new information system for housing building management. Proposed solution based on analysis and implementation of this solution.

Objective is to make functional design of information system, which will satisfy requirements of a building owner.

Klíčová slova

Informační systém, Analýza, Správa, Bytový dům.

Key words

Information System, Analysis, Management, Housing building.

Bibliografická citace

FOTIJEV, P. IS pro správu bytového domu. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 54 s. Vedoucí bakalářské práce Ing. Jan Luhan, Ph.D..

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů jsou úplné a že jsem ve své práci neporušil autorské práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

Petr Fotijev

V Brně dne 30. června 2013

.....

Podpis

Poděkování

Chtěl bych na tomto místě poděkovat především vedoucímu mé bakalářské práce Ing. Janu Luhanovi, Ph.D. za ochotu, trpělivost a cenné rady při psaní této práce. Také bych chtěl poděkovat mé rodině a přátelům za podporu během studia.

Obsah

Úvod	11
Cíl práce a vymezení problému	12
1 Teoretická východiska práce	13
1.1 Správa nemovitosti	13
1.2 Terminologie.....	13
1.3 Informační systém	14
1.3.1 Okolí informačního systému	14
1.3.2 Uživatelé IS	15
1.3.3 Veřejný IS.....	16
1.3.4 Interní IS	17
1.4 Doporučené vlastnosti informačních systémů	18
1.4.1 Jednoduchost	18
1.4.2 Odolnost	18
1.4.3 Stálost	18
1.4.4 Respektování zákonů a zvyklostí	19
1.5 Modely životního cyklu projektu IS/ICT	19
1.5.1 Vodopádový model	19
1.5.2 Spirálový model	21
1.5.3 Iterativní model	22
1.5.4 Další modely.....	22
1.6 W3C	22
1.6.1 Skupiny norem W3C	23
1.7 HTML 5	23
1.8 CSS 3	25

1.9	Architektura MVC	25
1.9.1	Model.....	26
1.9.2	View	26
1.9.3	Controler.....	26
2	Analýza problému a současné situace	27
2.1	Funkčnost IS	27
2.1.1	Požadavky	27
2.1.2	Specifikace	28
2.2	Use Case diagram	29
2.3	ER diagram	31
2.4	Plán vývoje	33
2.5	Databáze	33
2.5.1	Microsoft SQL Server 2012 R2 Express	34
2.5.2	MySQL Comunity Edition	34
2.5.3	PostgreSQL	34
2.6	Frameworky PHP.....	34
2.6.1	Nette Framework.....	35
2.6.2	Zend Framework	35
2.7	Preprocesory CSS	35
2.7.1	Sass.....	36
2.7.2	Less.....	36
2.8	Analýza konkurence	36
2.8.1	DES (Domovní Evidenční Systém).....	37
2.8.2	Windomy	37
2.8.3	nase-sprava-domu.cz	39

2.9	SWOT analýza.....	41
3	Vlastní návrhy řešení a jejich přínos	42
3.1	Cena systému	42
3.2	Zvolené technologie.....	42
3.3	Grafický návrh	43
3.4	Ukázky implementace	45
3.4.1	Nastavení routování.....	45
3.4.2	Nastavení zabezpečení	46
3.4.3	Lokalizace	46
3.4.4	Ukládání obrázků galerií	47
3.5	Praktické využití	48
4	Závěr.....	49
	Seznam použitých zdrojů	50
	Seznam obrázků	52
	Seznam tabulek.....	53
	Seznam příloh.....	54

Úvod

V dnešní době se majitelé domů potýkají s problémem správy svých nemovitostí. Hlavní problém nastává v komunikaci s nájemníky. Informování nájemníků o aktuálních věcech, evidování a řešení jejich přání, stížností, sledování výběru nájmů, jsou činnosti zabírající v souhrnu velké množství času. Podařilo se mi navázat spolupráci s majitelem domu, kterému nevyhovovalo žádné existující řešení na trhu, a proto by chtěl navrhnout a vytvořit nový informační systém na míru.

Cíl práce a vymezení problému

Hlavním cílem této práce bude vytvoření návrhu funkčního informačního systému podporujícího majitele bytových domů podle požadavků konkrétního majitele. Funkčnost ověřit implementací části návrhu s využitím skriptovacího jazyka PHP.

První, teoretická část, je zaměřena na problematiku vývoje informačních systémů, seznámení se s problematikou správy domu, způsoby řešení projektů vývoje software, standardizováním webových technologií a novými trendy.

V druhé části se budu věnovat analýze požadavků a vytvoření návrhu aplikace, analýze možných technologií vhodných pro použití v projektu a průzkumu konkurence a jejich řešení.

Závěr práce bude pojednávat o samotné implementaci, návrhu grafického rozhraní, navržení ceny a zhodnocení vytvořeného informačního systému.

1 Teoretická východiska práce

1.1 Správa nemovitosti

Správa nemovitosti zahrnuje činnosti, které umožňují nájemníkům bezproblémové využívání nemovitosti a odstranění případných překážek v chodu nemovitosti. Tyto činnosti zajišťuje přímo majitel nemovitosti nebo pověřená firma v plném nebo částečném rozsahu (1 str. 234-236).

Činnosti správy nemovitosti zahrnují:

- úklid nemovitosti,
- opravy závad,
- údržbu předcházející vzniku závad,
- plánování oprav a údržby (1 str. 234-236).

Správu nemovitosti provádí:

- majitel nemovitosti,
- bytové družstvo,
- společenství vlastníků,
- externí firmy zabývající se daným problémem (1 str. 234-236).

1.2 Terminologie

Terminologie informatických disciplín je definována podle normy ČSN 36 9001. Pojem informační technologie zahrnuje všechny způsoby využívány k pořízení, uchování, zpracování, prezentaci a přenosu dat. Komunikace je sdělování a přijímání informací vzájemně mezi lidmi, stroji i mezi strojem a člověkem (2 str. 30).

Používají se tyto zkratky:

- **IS** – informační systém (Information System)
- **IT** – informační technologie (Information Technology)
- **ICT** – informační a komunikační technologie (Information And Communication Technology)

- **IS/IT (IS/ICT** - Information System/Information And Communication Technology) zdůrazňuje provázanost informačních systémů na informačních technologiích a využívání informačních technologií k rozvoji informačních systémů (2 str. 30).

1.3 Informační systém

Systém je účelově definovaná množina prvků a vazeb mezi nimi. Systém vymezuje určitou část reality nebo část představy o realitě současné či budoucí. Rozsah systému je dán čistě užitkově, může představovat část reality či fikce, kterou jsme se rozhodli zkoumat nebo řešit (2 str. 45).

Informační systém je speciálním typem systému se vzájemně propojenými informacemi a procesy, které s těmito informacemi pracují tak, že vstupní informace transformují na výstupní informace. Dobře vymyšlený informační systém ve správných rukou se stává nástrojem veliké síly pro vykonávání i velmi náročných činností. Důležité je, aby integrace informačního systému činnosti usnadňovala, nepřidávala zbytečnou režii a neobtěžovala uživatele dalšími činnostmi, které před zavedením nebyly nutné. Každý kdo se podílí na tvorbě, realizaci a provozu informačních systémů, včetně uživatelů, by si měl tyto základní informace uvědomit a mít je na paměti (2 str. 44-45).

1.3.1 Okolí informačního systému

Okolí informačního systému tvoří jiné systémy a prvky s vazbami ke zkoumanému systému, které změnou svých vlastností ovlivňují systém nebo naopak mění své vlastnosti v závislosti na systému. Moderní informační systémy s okolím komunikují ve velkém rozsahu (např. získání aktuálního měnového kurzu od České národní banky, platba kartou). Činnost systému je tedy značně ovlivňována okolím, proto je nutné pro úspěšný návrh a provoz projektu informačního systému nezbytné určit vazby okolí k systému a podrobně je rozebrat (2 str. 46-47).

Prvky systému pro komunikaci s okolím se nazývají hraniční prvky. Typickým příkladem je přepážková pracovnice v bance nebo telefon při objednání večeře.

Pro uživatele hraniční prvky představují nejen prostředek pro komunikaci se systémem, ale také reprezentaci systému jako celku a mohou působit na uživatele přitažlivě ale i odpudivě. Důležité je věnovat se návrhu těchto prvků s pečlivostí, vytvořit lákavý design a odladit chyby, protože hraniční prvky mají význam nejen technický ale i marketingový a ekonomický. Z aktivit uživatelů při využívání hraničních prvků lze pomocí analýzy získat podklady pro marketingové a propagační činnosti (2 str. 46-47).

1.3.2 Uživatelé IS

Uživatel využívá služeb poskytovaných systémem. Pro řešení systému je důležité, zda je uživatel zařazen do systému nebo do okolí (2 str. 47).

- Uživatel jako prvek okolí – vazby s okolím a ostatními uživateli nejsou podstatné a nelze mu nic přikazovat ani zakazovat. Uživatel je omezován pravidly, jejichž porušení může být trestáno různými způsoby. Všechny tresty ale pouze omezují využití služeb systému v časovém horizontu nebo trvale. Aby bylo možné systém co nejlépe navrhnout, je nutné analyzovat škálu možností chování uživatele a to včetně extrémů. Pokud je systém zaměřen na konkrétní skupinu uživatelů je možné vytvářet určité předpoklady o uživateli. Obecně od uživatele nelze nic očekávat a nic předpokládat (2 str. 47-48).
- Uživatel jako prvek systému – vazby s ostatními uživateli jsou důležité a jsou předmětem řešení, řízení a organizování. Uživatel má pevně daný pracovní řád, jeho porušení lze trestat a sankcionovat i mimo rozsah systému například finanční pokuta (2 str. 47).

Tabulka 1: Uživatelé systému
(Zdroj: 2 str. 48)

Uživatel	Uživatel je prvkem	
	systému	okolí
vazby k prvkům systému	přímé	prostřednictvím hraničního prvku
vazba na jiné uživatele	musí nás zajímat, vazbu analyzujeme a řešíme	nezajímá nás, nemáme na ni vliv
vlastnosti, znalosti, dovednosti, komunikativnost	určujeme a předepisujeme	odhadujeme, předpokládáme (extrémy, ne průměr)
chování v různých situacích	určujeme a předepisujeme (v rámci řešení systému)	předpokládáme v širokém rozmezí možností
časové podmínky styku se systémem	určíme při návrhu organizace systému	stanovíme podle marketingového odhadu (co nejvolnější)
činnosti při styku se systémem	detailně a přesně určujeme v pracovních postupech a předpisech	stanovíme co nejjednodušší pravidla a návody
řízení činností	uživatel je řízen jako jiné prvky systému	uživatel může dostávat pokyny, ale nemusí je uposlechnout
pracovní postupy	pevně stanoveny při řešení i za provozu	dány pouze návodem, nemusí být dodrženy
nedovolené/nesprávné činnosti	nepřipouštět se	musí být předpokládány
sankce při nedovolené/nesprávné činnosti	postih (kázeňský, finanční, ...)	upozornění, zdvořilé napomenutí, odepření funkce systému
skupiny uživatelů	v každé skupině stejná pravidla, předpisy sankce atd.	každý uživatel může mít jiné vlastnosti a chování

1.3.3 Veřejný IS

Veřejné informační systémy fungují na principu, kde jeden právní subjekt umožňuje ostatním subjektům využívat svých služeb nebo se jiným způsobem podílet na provozu služby, například poskytováním dat. Veřejné systémy poskytují data zaměřená na specifické oblasti uživatelům, kteří data z této oblasti potřebují pro svou činnost. Práva na užití těchto informací nepatří uživatelům, proto může být přístup zpoplatněn a nemusí být umožněn všem. Přístup potom

uživatel může získat formou registrace, licence, členstvím ve skupině, příslušností ve státní složce a dalšími. Systém může být distribuován použitím pevných medií, downloadu nebo online spojením (2 str. 50-51).

Principy řešení a řízení veřejných informačních systémů:

- obchod s informacemi - encyklopedie
- povinnost předávání informací – stát informuje občany
- nutnost práce s externími informacemi – systémy koordinace rádiového kmitočtového spektra
- potřebnost, účelnost a užitečnost – internet, databáze získaných zkušeností (2 str. 51)

Uživatel ve styku s informačním systémem vystupuje pouze jako externí subjekt v okolí systému. Vlastník informačního systému zodpovídá za bezpečnost dat, správu a provoz systému. Tyto úkony zabezpečuje sám nebo je může outsourcovat. Poslední dobou dochází k propojování systémů v národní i nadnárodní rovině (2 str. 51).

1.3.4 Interní IS

Jsou systémy podporující činnost právnické osoby od malých podniků s několika zaměstnanci až po mezinárodní korporace. Hlavním účelem je poskytovat relativní, aktuální a přesná data ve stanovených termínech a podobě vhodné pro podporu rozhodování. Data vznikají jako produkt práce firmy, jako výsledek zákonné povinnosti, ale také jako podpora plnění cílů firmy, jsou majetkem firmy. Pokud firma používá externí data, je povinna získat je legální cestou a za vlastnictví práv je odpovědná. Architektura informačního systému je dána podnikatelskou strategií a oborem podnikání. Uživatel je součástí systému, podle své role v podniku má omezen přístup k informacím a je zapojen do procesu zabezpečení. Správu, provoz a rozvoj informačního systému může provádět firma ve vlastní režii nebo pomocí outsourcingu v podobě vyhovující jejím požadavkům jako bezpečnost, spolehlivost a dostupnost (2 str. 57-58).

1.4 Doporučené vlastnosti informačních systémů

Tyto vlastnosti by měly obecně platit pro veřejné informační systémy, avšak jejich dodržení při vývoji interního IS umožní vytvořit kvalitní aplikaci se standardním chováním, na které je uživatel zvyklý ze svého každodenního styku s informačními technologiemi. Samozřejmě ne vždy jsou tyto zásady dodrženy (2 str. 52).

1.4.1 Jednoduchost

Přestože systém není jednoduchý, měl by se tak jevit uživateli. Obsluha IS by měla být co nejjednodušší s logickým uspořádáním ovládacích prvků. Uživateli musí být srozumitelné vstupy, výstupy a komunikace se systémem z jeho pohledu a ne z pohledu informatika. I když je možné do jisté míry předpovědět modelového uživatele, systém by měl být navržen tak, aby vycházel vstříc uživatelům s mírně podprůměrným IQ a způsobem uvažování. Nedodržení této vlastnosti má za následek zvýšení dotazů na pasivní či aktivní centra nápovědy a snížení efektivnosti informačního systému (2 str. 52).

1.4.2 Odolnost

Odolnost vůči neznalosti, zlomyslnosti, zločinnosti uživatelů, chybám, škodám a ztrátám z technických důvodů je pro systém a informace v něm uložené kritická. Tyto jevy je nutné brát na zřetel již při návrhu a předcházet jejich vzniku a vytvářet ochranná opatření pro případ, že se je nepodaří předpovědět nebo ani nemohou být očekávána (2 str. 52).

1.4.3 Stálost

Systém by měl dodržovat zavedené postupy, strukturu menu, formát vstupních a výstupních informací, číselníky a další. Ovšem rozvíjení systému a implementování nových technologií toto nabourává, je nutné najít rovnováhu mezi vývojem a stálostí systému (2 str. 52).

1.4.4 Respektování zákonů a zvyklostí

Systém by měl respektovat právní normy, etické normy a zvyklosti i neoficiální nepsané i psané zákony (2 str. 53).

1.5 Modely životního cyklu projektu IS/ICT

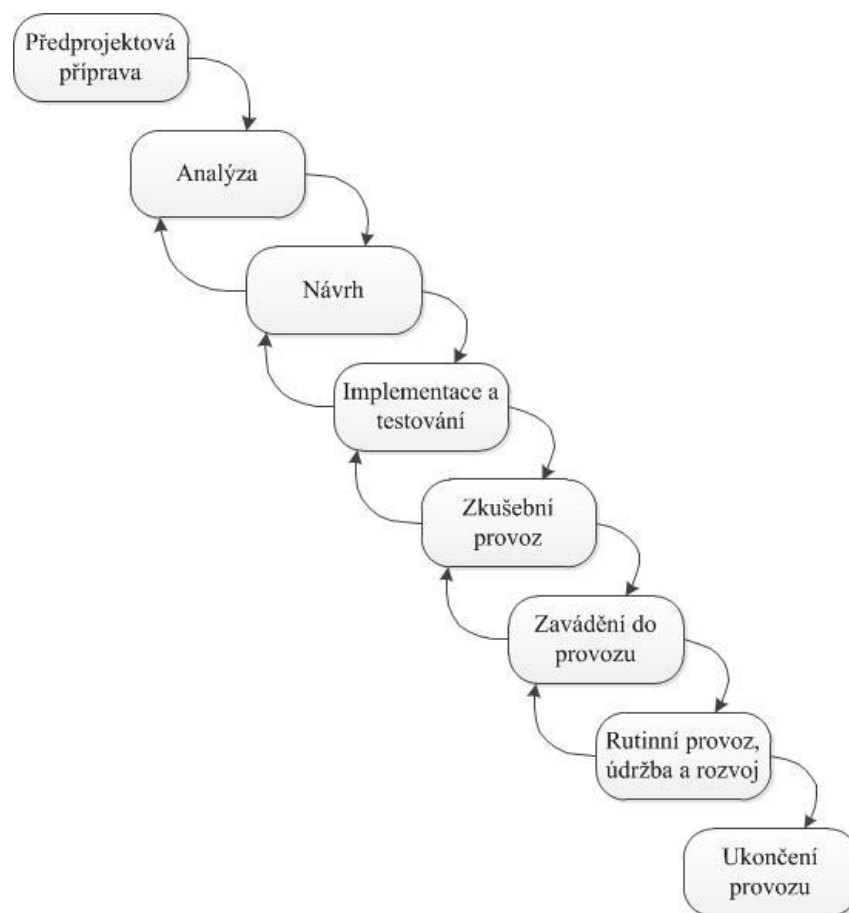
Modely vývoje se přizpůsobovaly podmínkám na trhu s informačními systémy a používaným informačním technologiím. Starší modely spíše popisují ideový přístup k projektu jako celku. Díky své abstrakci se používají dodnes, ale pro taktické a strategické řízení projektu IS/ICT. Moderní modely jsou zpracovány do detailů, pro operativní řízení se používají s podporou informačních technologií (3 str. 68).

1.5.1 Vodopádový model

Vodopádový model patří mezi nejstarší modely. Vychází z doby, kdy poptávka na trhu projekčních prací převyšuje nabídku a každý projekt informačního systému se stává jedinečným dílem na zakázku (3 str. 69).

Předpoklady pro nasazení vodopádového modelu životního cyklu projektu jsou následující:

- jednotlivé etapy jsou vykonávány postupně po ukončení etapy předchozí. Výsledky etapy se po ukončení vyhodnotí a poslouží jako podklady pro následující etapu,
- nejlepší výsledek se dosáhne při dodržení pořadí na obrázku 1 (3 str. 52).



Obrázek 1: Vodopádový model životního cyklu projektu

(Zdroj: 3, str. 70)

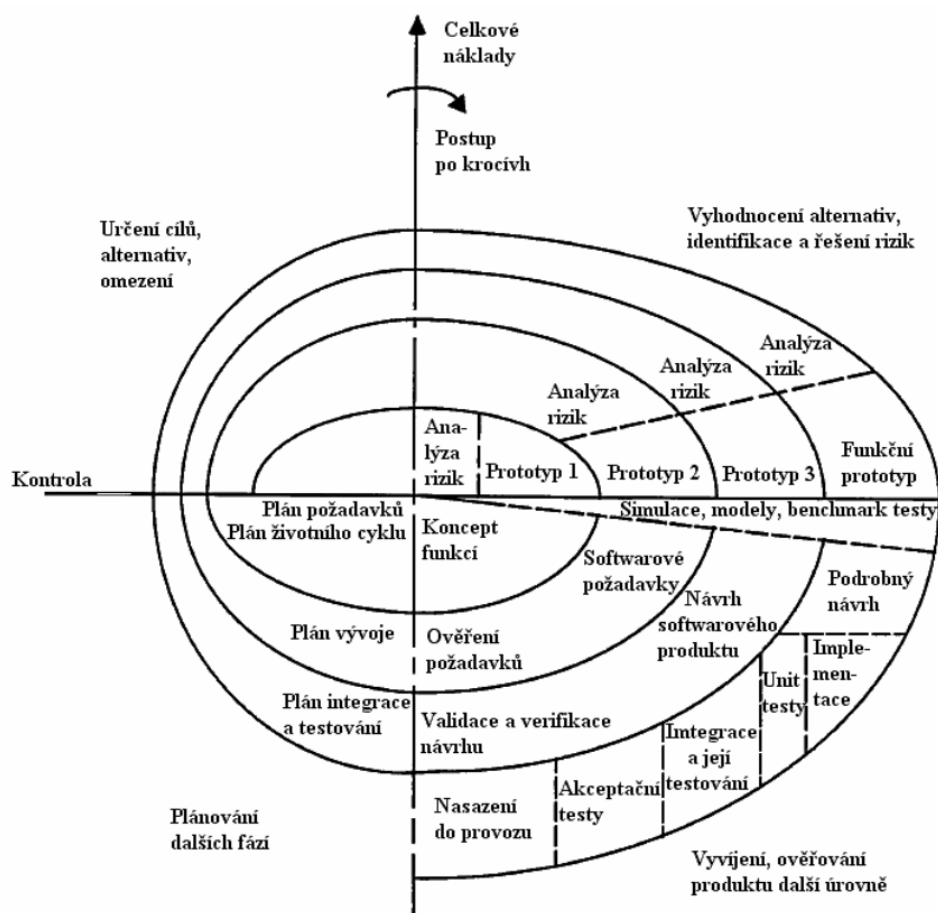
Pro ekonomické vztahy mezi dodavatelem projektu a jeho objednatelem platí:

- objednatel projektu hradí veškeré finanční náklady na tvorbu projektu informačního systému,
- riziko úspěšnosti projektu leží objednateli,
- náklady dodavatele zahrnují vyškolení vlastních zaměstnanců a pořízení informačních technologií spojených s vývojem (3 str. 70).

Díky svému charakteru, jednoduchosti, srozumitelnosti, jednoduchou transformací do etap a fází do časové osy apod., se tento model používá převážně pro řízení projektu IS/ICT a sledování jeho průběhu. Model je však stále možné vidět na projektech některých manažerských aplikací. Díky ekonomickým vztahům mezi dodavatelem a objednatelem je vhodný pro firmy snažící se proniknout na trh s novým řešením (3 str. 71).

1.5.2 Spirálový model

Tento systém vznikl v době, kdy na trhu došlo k vyrovnání mezi poptávkou a nabídkou projektů IS/ICT. Proto dodavatel začíná vyhledávat části trhu, kde existuje možnost uplatnění podpory IS/ICT. Snaží se zjistit velikost poptávky po informačním systému nebo jeho části. V případě, že zjistí možnost uplatnění v tomto odvětví trhu, vytvoří první prototyp. Ten nabídne na trhu, a pokud je prototyp kvalitní získá prvního zájemce (3 str. 71-73; 4 str. 108).



Obrázek 2: Spirálový model

(Převzato z: <http://testovanisoftware.cz/wp-content/uploads/2011/07/spirala.png>)

Takto vznikne další prototyp obsahující požadavky prvního objednatele, ze kterých budou některé funkčnosti, které jsou nedostatkem prvního prototypu, vybrány a použity pro ostatní. Za tyto funkčnosti zaplatí objednatel méně než za funkčnosti specifické pro něj. Tento model tak vyrovnává náklady mezi objednatelem a dodavatelem (3 str. 72-73; 4 str. 108).

1.5.3 Iterativní model

Iterativní model vychází ze spirálového modelu, na který implementuje vodopádový model. Hlavní myšlenkou tohoto modelu je, že člověk dokáže řešit více malých problémů než jeden velký. Projekt se tedy rozdělí do několika menších logických částí, které na sebe navazují, tím vzniká řetěz menších vodopádů. Jednotlivé iterace jsou testovány a jejich výstupem by měla být funkční část systému. Tento přístup snižuje riziko nespokojenosti zákazníka, jelikož může pokaždé iteraci provést kontrolu a podat zpětnou vazbu. Na rozdíl od projektu řešeného s použitím vodopádového modelu dokáže flexibilně reagovat na změny návrhu a přidávání nových funkcí (4 str. 109).

1.5.4 Další modely

Novější modely jsou komplexnější a využívají podporu informačních technologií, patří mezi ně Síťový model, agilní modely nebo MDA (Model Driven Architecture).

1.6 W3C

Protože navrhovaný systém bude v podobě webové prezentace, je důležité zjistit, kdo se zabývá udržováním norem na tomto médiu.

World Wide Web Consortium (W3C) je mezinárodní organizace spolupracující s odbornou veřejností na rozvoji a standardizování webových stránek. Cílem konsorcia je „Rozvíjet World Wide Web do jeho plného potenciálu vývojem protokolů a směrnic, které zajistí dlouhodobý růst Webu“(5). Konsorcium vede zakladatel World Wide Webu Tim Berners-Lee, který je také jedním z autorů specifikací URL, HTTP a HTML, základních kamenů webu (5).

V současné situaci dodržování norem webovými vývojáři nezaručuje bezchybné fungování vytvořených dokumentů z důvodů nedostatečné podpory výrobců aplikačního softwaru, zejména webových prohlížečů a serverů. Nové normy jsou zpravidla implementovány se zpožděním a to i v řádu několika let. Proto je nutné před začátkem projektu prozkoumat stav dodržování norem a implementaci

nových technologií v běžně používaném softwarovém vybavení uživatelů a provozovatelů budoucí aplikace (6).

1.6.1 Skupiny norem W3C

- **HTML (Hypertext Markup Language)** v roce 1991 ho navrhl Tim Berners-Lee, jako jednodušší formu definice dokumentů oproti v té době používaných TeX, PostScript a SGML. Aktuální doporučovaná verze je HTML 4.1 z roku 1999 (5; 6).
- **XHTML (Extended Hypertext Markup Language)** bylo navrženo v roce 2000 za účelem aplikace striktnějších pravidel XML na HTML 4.1, jehož syntaxi kopíruje nebo rozšiřuje. V roce 2001 je rozšířen o modularizaci a povýšen na verzi 1.1, která je doporučována standardem do současnosti.
- **CSS (Cascading Style Sheets)** je jazyk pro definici stylu zobrazení dokumentů HTML, XHTML a XML, který navrhl Håkon Wium Lie. Od roku 2011 je konsorciem W3C doporučená verze 2.1 (5;6).
- **XML (eXtensible Markup Language)** není přímo jazyk, spíš soubor pravidel, podle kterého lze navrhovat jazyky k libovolným účelům. Verze 1.0 vznikla v roce 1999. Doporučené je používat pátou edici této verze z roku 2008 (5; 6).
- **PNG (Portable Network Graphics)** je rozšířený souborový formát pro bezztrátově komprimované ukládání rastrových obrázků. Vznikl jako náhrada za patentově chráněný GIF (Graphics Interchange Format) v roce 1996. Může také nahrazovat běžné užití TIFF (Tag Image File Format). Formát udržuje také informace o alfa kanálu, čím umožňuje průhlednost (5;6).

1.7 HTML 5

HTML 4.1 a XHTML 1.1 bylo konsorciem W3C doporučeno před 14 respektive 12 lety. Z toho důvodu se dnes standartní webové aplikace potýkají s omezeními jazyka a prohlížečů. Vývoj specifikace HTML 5 začal v roce 2004, přičemž se počítá s plným uvedením do provozu v roce 2014. Na vývoji se podílí i velcí hráči ve vývoji aplikací včetně Apple a Microsoft. Nové HTML bude obsahovat sadu

vlastností, které budou vykonávat funkce prováděné v dnešní době pomocí externích technologií jako Flash a JavaScript. Specifikace bude myslet i na funkce spojené s mobilními zařízeními jako geolokace a služby založené na poloze. Tyto funkce umožní bohatší web bez nutnosti použití rozšíření webového prohlížeče (7).

Podpora HTML 5 je už dnes pomalu integrována do hlavních webových prohlížečů. Microsoft počítá s podporou HTML 5 v Internet Exploreru od verze 10. S novou technologií začali také experimentovat a připravovat se na její integraci do webové prezentace některé nejnavštěvovanější weby, s Youtube v čele. Na adrese www.youtube.com/html5 je verze používající nové elementy, funkce a pravidla (7).

Mezi nové vlastnosti HTML5 patří:

- Canvas, nový element, který umožňuje vývojáři kreslit a zobrazovat grafiku, video a animace, hlavně pomocí JavaScriptu. Díky tomu lze renderovat grafiku na straně klienta a tím snížit objem přenesených dat a zrychlit renderování stránek s velkým obsahem grafiky.
- Video element umožní uživatelům přehrávání videa bez nutnosti instalace odpovídajícího přehrávače. K tomu se používají nechráněné video formáty jako Ogg, Theora a VP8. HTML značka tedy říká prohlížeči, že se k následující informaci má přistupovat jako k HTML5 kompatibilnímu video streamu.
- Podpora služeb závislých na lokaci pomocí nového rozhraní.
- Možnosti práce offline by se měly podstatně rozšířit díky ukládání dat a obslužného kódu do AppCache, tím umožnit webovým aplikacím fungovat lokálně bez nutnosti připojení k internetu. HTML5 nabízí další nástroje pro výrobu offline webových aplikací jako klientská SQL databáze a cachování dat.
- Web Workers element umožňuje provádění skriptu na pozadí, který nebude přerušován jinými skripty a interakcí uživatele.

- HTML5 umožňuje zápis jak v klasické HTML podobě, tak i v XML. Stránky psané v XML jsou sice složitější na vývoj pro programátora kvůli své striktnosti, a díky jí se stránky zobrazují rychleji (5; 7).

1.8 CSS 3

Konsorcium W3c zahájilo vývoj třetí verze CSS v roce 2005 přičemž dokončení se předpokládá v roce 2015. CSS 3 přináší zjednodušení vytváření běžných prvků grafického návrhu. Ovšem již v dnešní době je většina vlastností podporována nejpoužívanějšími webovými prohlížeči (5).

CSS3 nabízí nástroje pro jednoduchou práci s okraji, díky kterým odpadá nutnost řešit tyto části webové prezentace pomocí obrázků, tedy se snižuje i množství přenášených dat. Pomocí *border-radius* lze zaoblovat rohy blokových prvků. Vlastnost *box-shadow* umožňuje vytvářet stíny (8).

Nové vlastnosti se týkají i práce s pozadím prvků. *Background-size* umožňuje nastavit velikost obrázku použitého v pozadí. Nově je možné nastavit i více pozadí jednomu prvku (8).

Barvy lze nyní definovat i s použitím barevného modelu HSL. Vlastnosti *opacity* umožňuje nastavit průhlednost bloku. O průhlednost je také rozšířen původní model RGB i nový HSL (8).

Mezi nové vlastnosti textu patří přidání stínování pomocí *text-shadow*. Další velmi užitečnou vlastností je upravení zalamování dlouhých slov (*word-wrap*), které mohou být automaticky rozdělena CSS. Závěrem bych zmínil vlastnost *text-overflow*, umožňující zobrazující znaků napovídajících, že došlo k přetečení textu, jako jsou tři tečky (8).

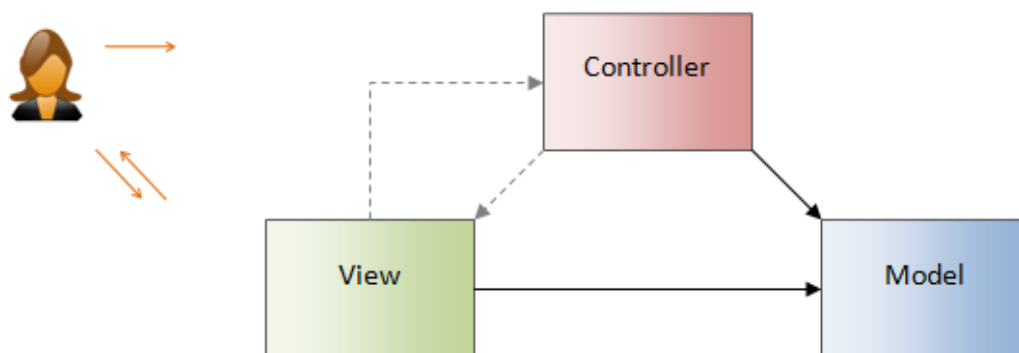
1.9 Architektura MVC

Architektura Model-View-Controller se dnes používá ve většině běžně používaných frameworků, proto je vhodné se s touto architekturou seznámit. Architektura MVC vznikla z potřeby oddělit kód obsluhy (controller), aplikační

logiku (model) a zobrazení dat (view). Toto rozdělení kódu aplikace zpřehledňuje, ulehčuje vývoj a testování jednotlivých částí (9).

1.9.1 Model

Model je vrstva mezi aplikací a fyzicky uloženými daty, obsahující aplikační logiku jako akce přihlášení uživatele, přidání položky do košíku atd. Model si udržuje a spravuje svůj vnitřní stav a s okolím komunikuje pomocí pevně daného rozhraní, přičemž se o ostatní části architektury MVC nezajímá (9).



Obrázek 3: Interakce uživatele s architekturou MVC

(Zdroj: <http://www.zdrojak.cz/wp-content/uploads/2009/05/03-MVC-interakce-s-uzivatelem-124164094993085.png>)

1.9.2 View

View je vrstva aplikace starající se komunikaci s uživatelem a zobrazení dat uživateli. Tato vrstva zobrazuje data modelu je ale ovládána controlerem, kterému předává akce uživatele (9).

1.9.3 Controler

Controler je v podstatě řídicí jednotkou. Na základě akce uživatele rozhoduje, jaká bude reakce systému a jaká data, získané od modelu, se mají zobrazit v pohledu (9).

Příklad přidání zboží do košíku. Kliknutí uživatele na přidání zboží do košíku způsobí předání informace z pohledu controleru, který řekne modelu, aby do košíku přidal vybrané zboží, poté nařídí pohledu, aby aktualizoval stav košíku.

2 Analýza problému a současné situace

Tato část se zabývá analýzou požadavků zadavatele a jejich transformací na specifikaci systému. Dále vytvořením podkladů pro vývoj jako je Use Case diagram, ER diagram a vytvoření plánu vývoje softwaru. Nezbytná je i analýza možných technologií vhodných k vývoji projektu. Závěr této kapitoly se zabývá konkurencí a jejími produkty.

2.1 Funkčnost IS

Zadavatel je majitelem bytového domu v centru města Brna. V domě je přes 20 pronajímaných bytů. V době sestavení požadavků bydlí ve svém pronajímaném domě, ale v časovém rozmezí půl roku se plánuje přestěhovat mimo Brno. S tím se mu zvětšuje režie těchto jednání o cestu do Brna a zpět.

Zadavatel odhaduje, že stráví činností týkající se správou domu, komunikaci s nájemníky a realitními kanceláři 7 až 10 hodin týdně. Přičemž si hodiny svého času cení na tisíc korun.

Do každého pronajímaného bytu je zaveden internet. Majitel se zaměřuje na pronájem bytů mladým lidem, u kterých předpokládá vlastnictví zařízení s možností přístupu na internet.

2.1.1 Požadavky

Zadavatel by si přál, aby systém běžel na internetu a tak byl přístupný odkudkoliv, ale pouze pověřeným lidem. Dále by si přál, aby mu usnadnil komunikaci s realitními kanceláři, které využívá k zprostředkování pronájmů. Informační systém by měl podporovat komunikaci s nájemníky a informovat je o aktuálním dění v domě, stavu jejich nájmu a výši vyúčtování. Aplikace by měla obsahovat soubor návodů k vybavení bytu a umožňovat správu dokumentů a smluv. Měl by obsahovat kalendář, do kterého budou moci osoby obstarávající úklid domu a opravy výtahu zadávat své úkony. Majitel by si přál umožnit nájemníkům komunikovat mezi sebou a podávat případné návrhy na zlepšení. Jelikož v domě bydlí také cizinci, zadavatel žádá možnost jazykových mutací.

2.1.2 Specifikace

Na základě požadavků jsem vytvořil následující specifikaci informačního systému.

Informační systém by měl sloužit pouze pověřeným osobám a nepočítá se s webovou prezentací domu. IS by měl obsahovat anglickou mutaci a do budoucna by měl být rozšířen o španělštinu a němčinu. Počítá se následujícími rolemi uživatelů:

Administrátor (majitel) - získává informace, spravuje systém, informuje nájemníky

Nájemník - bydlí v domě, získává aktuality a informace o provozu domu

Realitní kancelář - přístup realitních kanceláří, které zprostředkovávají klienty pro volné byty, získává informace, které byty jsou volné a informace o nich

Uklízeč - zadává do kalendáře co se uklízelo

Opravář – informuje, kdy se bude provádět oprava a kontrola výtahu

Souhrn funkcí uživatelské části:

- informace o všech bytech včetně galerie
- dokumentace k zařízení bytu a obecně návody na připojení k internetu a další ve formátu PDF
- informace o zaplaceném nájemném pro každého konkrétního nájemníka
- zobrazení vyúčtování za spotřebovaný plyn, elektřinu, teplo a vodu zatím v podobě PDF ke stažení
- kalendář obsahující informace o úklidu a opravách v domě
- aktuální informace týkající se nájemníků a domu
- kontaktní formulář a kontaktní informace majitele
- zprávy uživatelům
- jednoduchá moderovaná diskuze
- editace vlastních osobních informací
- zobrazení internetového provozu

administrační rozhraní bude obsahovat následující:

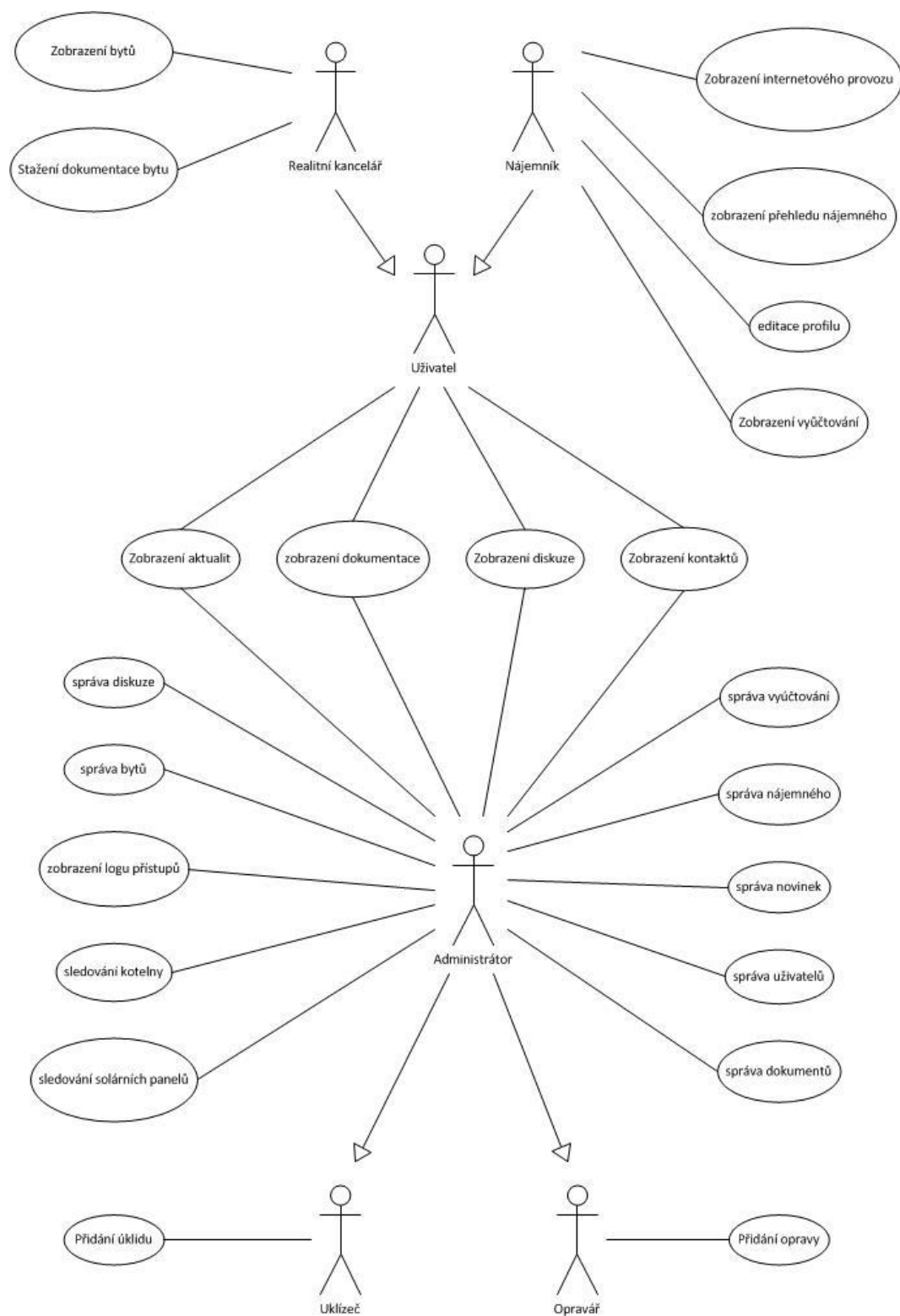
- rozhraní pro zadávání úklidu a oprav
- administrace novinek
- informace o bytech
- budoucí integrace sledování solárních panelů a kotelny
- zasílání správ uživatelům
- rozhraní pro výpočet vyúčtování
- správa nájemného (ruční a automatické z uploadovaného výpisu)
- správa uživatelů
- uploadování informací o internetovém provozu
- logování přístupů do systému

Aktuality, dokumentace, diskuze a kontakt jsou přístupné všem rolím. Všichni uživatelé mají přístupné rozhraní zpráv, o kterých jsou informováni na e-mailu. Přístup pro realitní kanceláře umožní pouze prohlížet informace o jednotlivých bytech. Nájemník má přístup k informacím o historii zaplaceného nájemného, vyúčtování za energie, přehled internetového provozu a editaci svých údajů.

Do budoucna se počítá s vytvořením výpočetní logiky pro vyúčtování energií a integrací rozhraní sledování solárních panelů a kotelny.

2.2 Use Case diagram

Specifikované požadavky jsem zobrazil do diagramu užití, který slouží jako grafické zobrazení funkčnosti systému. Díky své přehlednosti a názornosti se diagram užití hodí jako podklad zobrazující budoucí systém zadavateli. Po schválení návrhu slouží k identifikaci operací, které je nutné implementovat.



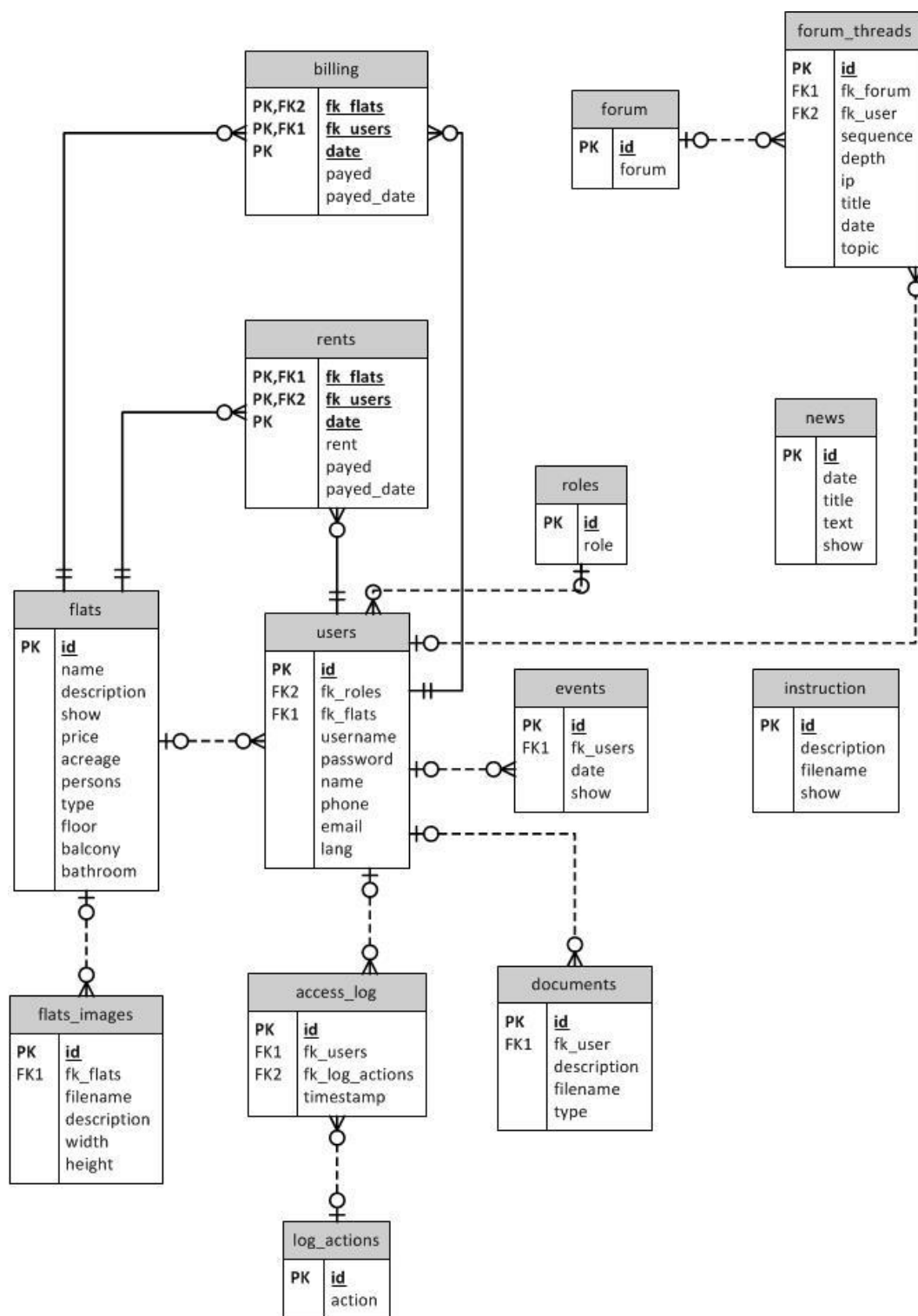
Obrázek 4: Use Case diagram
(Zdroj: Vlastní)

Na diagramu jsou zobrazeny role Realitní kancelář a Nájemník, které rozšiřují obecnou roli Uživatel. Realitní kanceláři patří činnosti spojené s prohlížením bytů a získání jejich dokumentace. Akce role Nájemník zahrnují kontrolu stavu jeho s majitelem domu včetně zobrazení soukromých dokumentů. Role Administrátor má společné případy užití s obecnou rolí Uživatel, další její funkčnosti se zabývají správou systému a informováním ostatních rolí. Role Uklízeč a Opravář mají své činnosti zahrnující vkládání událostí do kalendáře a jsou rozšířeny Administrátorem, který má k jejich činnostem také přístup. Všechny role, mimo Uklízeče a Opraváře, mají přístup k internímu systému zpráv. Z diagramu je vidět, že pouze administrátor může rozesílat hromadné zprávy.

2.3 ER diagram

ER diagram se používá k popisu informací, se kterými bude systém pracovat a vztahů mezi nimi. Na základě tohoto diagramu se posléze vytvoří struktura databáze.

Hlavní entitou je flats obsahující záznamy o jednotlivých bytech. Na ty jsou navázány obrázky a entity vztahů s uživateli, billing a rents. Logování přístupů probíhá nad entitou access_log. Funkčnosti fóra se týkají entity forum a forum_threads. Na uživateli je dále závislá entita dokumentů. Návrh obsahuje i nezávislé entity jako novinky a dokumenty návodů.



Obrázek 5: ER diagram
(Zdroj: Vlastní)

2.4 Plán vývoje

Plán vývoje vychází z iterativního modelu, kdy jednotlivé fáze jsou po dokončení testovány. Před dokončením fáze probíhá interní testování, z důvodu vyhnutí se trapných chyb při předvedení zadavateli. Po dokončení třetí fáze a následujících jsou předvedeny změny zadavateli, aby mohl případné nedostatky rozporovat nebo dále specifikovat danou část. Po opravení nedostatků a případných úpravách nastává vývoj následující fáze. Jazykové mutace bude provádět externí překladatel. K překladu dostane soubor obsahující identifikátory textů s jejich českou podobou. Po integraci předkladů je nutné projít celý systém a zkontrolovat, že změnou jazykové mutace nedošlo k přetečení textů.

Tabulka 2: Fáze vývoje
(Zdroj: Vlastní)

Fáze	Náplň
1.	Nastavení frameworku a databáze, inicializace technologií lokalizace, preprocesoru CSS, slučování CSS a JavaSriptu, nastavení routování
2.	Nastavení zabezpečení aplikace, vytvoření uživatelů a jejich rolí
3.	Funkčnost realitní kanceláře: přehled bytů včetně galerie a administrační části
4.	Společná část: aktuality, dokumenty, diskuze, kontakt a jejich administrační část
5.	Funkčnost nájemníků: zobrazení internetového provozu, přehledu nájemného, vyúčtování a editace vlastního profilu včetně administrační části
6.	Dokončení administrační části: logování přístupů, rozhraní pro úklid a opravy, správa uživatelů a jejich oprávnění, rozhraní pro zasílání a oznamování zpráv
7.	Integrace sledování solárních panelů a kotelny
8.	Jazykové mutace systému a testování

2.5 Databáze

Výběr vhodné databáze pro projekt vyžaduje zvážení mnoha faktorů. Pro mé řešení jsem se rozhodl, z důvodu poměrně malé databáze, zaměřit na zdarma dostupné produkty. Také by bylo vhodné, kdyby databáze byla bez problému dostupná na českých hostinzích.

2.5.1 Microsoft SQL Server 2012 R2 Express

MS SQL Server Express je zdarma dostupná databáze zaměřená na využití malými řešeními. Oproti placené verzi má několik omezení, znemožňující její použití velkými projekty. Maximální velikost každé databáze je omezena na 10 GB včetně logů. Hardwarově je omezena na využití pouze jednoho fyzického procesoru a maximální využití paměti RAM do velikosti 1 GB. Databáze musí běžet na serveru s platformou MS Windows. Databáze se instaluje s uživatelským rozhraním (10).

2.5.2 MySQL Comunity Edition

MySQL je k dispozici pod bezplatnou GPL licencí. Databázový systém je možné nainstalovat na různé operační systémy v čele s MS Windows a Linux. MySQL bylo původně navrženo pro rychlost a proto původně nepodporovalo trigger, pohledy ani uložené procedury. Tato rozšíření byla dodatečně přidána kvůli jejich nutnosti k výrobě moderních aplikací a ulehčení práce. MySQL nabízí mnoho typů úložných enginů. Tato databáze je velmi rozšířená na hostinzích, často se vyskytuje ve spojení s operačním systémem Linux, serverem Apache, a skriptovacím jazykem PHP. (11).

2.5.3 PostgreSQL

PostgreSQL je open source objektově relační databáze. Databáze běží na všech běžně používaných systémech jako Linux, Unix a Windows. Databáze umožňuje vývojářům vytvářet vlastní datové typy jako ISBN/ISSN čísla, IP adresy nebo Geographic Information System (GIS). Slabinou databáze jsou velké datové sklady, kdy se výkonem nevyrovná placeným řešením. Většinou bývá na hostinzích jako alternativa k MySQL (12).

2.6 Frameworky PHP

Framework je soubor nástrojů nabízející řešení opakujících se úloh např. spouštění aplikace, přihlášení, odeslání formuláře atd. Tyto akce jsou hotové nebo potřebují pouze nastavit a vývojář se tak může zabývat pouze řešením zadání.

2.6.1 Nette Framework

Je český open source framework napsaný Davidem Grudlem nabízený pod BSD licenci a GNU GPL licenci. K velkým kladům tohoto frameworku patří velká česká komunita vytvářející různá rozšíření a pluginy. Od verze 2 běží Nette pod PHP 5.3 a využívá jmenných prostorů. Mezi nevýhody Nette patří nezachování zpětné kompatibility v nových verzích. Controller MVC architektury se v Nette nazývá Presenter (9).

Nette nabízí množství ladících nástrojů jako Laděnka, propojení s FireBudem a FireLoggerem, měření časové náročnosti operací, vizualizaci chyb a vyjímek, Debugger Bar, logování chyb a přepínání mezi produkčním a vývojovým režimem. Debugger Bar nabízí možnost rozšíření existujícími nebo vlastními doplňky. K dalším silným stránkám frameworku patří práce s formuláři včetně validace a zabezpečení proti útokům Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), URL a UTF-8 attack. Na rozdíl od ostatních frameworků umožňuje obousměrné routování, díky kterému je možné odkazovat přes presentery a nikoliv pomocí pevně dané URL. Samozřejmostí je podpora pěkných URL (Cool URL) (9).

2.6.2 Zend Framework

Zend Framework je open source projekt pod licenci New BSD. Všechny knihovny a součásti jsou plně objektově orientované a vyhovují direktivě E_STRICT, vyžadující dodržování pravidel čistého kódu PHP. Velkou výhodou je modulárnost celého frameworku, umožňující používání pouze požadovaných částí. Kolem frameworku je obrovská komunita, podstatně větší než okolo Nette. Zend podporuje řadu technologií třetích stran (13).

2.7 Preprocesory CSS

CSS jako čistě deklarativní jazyk se hodí k popisu stylů webových stránek, ale s růstem webové prezentace také úměrně roste soubor jejich CSS stylů. S velikostí roste i složitost udržování stylů. Tento problém lze napravit psaním složitých komentářů a dokumentace nebo použitím preprocesorů CSS. Preprocesory jsou

v podstatě překladače zápisu CSS, přináší tedy do CSS něco z imperativního a objektového programování. Zápis CSS rozšiřují o proměnné, výpočetní výrazy, funkce, vnořování pravidel, zapouzdření a další funkce. Použitím preprocesorů vzniká čitelnější, přehlednější zápis a rapidně se zvyšuje znouvupoužitelnost.

S použitím preprocesorů CSS vzniká problém kdy a jak překládat zapsané CSS, protože webový prohlížeč mu nerozumí. Překlad může obstarávat JavaScript, démon sledující změny souboru nebo se musí provádět ručně (14).

2.7.1 Sass

Vytvořen v komunitě okolo jazyka HAML (jazyk pro zjednodušený zápis HTML) v roce 2007. Syntaxi stylů zprostil o zápis složených závorek a středníků, přičemž nahradil tuto logiku odsazováním na nové řádky. Tento zápis však ne každému vyhovoval. Proto vznikla verze Scss, ve které je zachována původní syntaxe CSS. Překlad kromě normálního způsobu, kdy je využit příkazový řádek, může zajišťovat démon, sledující změny v souborech na určených cestách (14; 15).

2.7.2 Less

Zápis vychází z čistého CSS. Původní implementace je v jazyku Ruby, ale existuje i předělávka v JavaScriptu pod názvem less.js a PHP implementace nazvaná LESS PHP. Implementace v PHP je velkou výhodou protože není třeba instalovat na server žádné další technologie, což může být v případě, kdy se nejedná o vlastní hosting, ale pronajatou službu, velmi komplikované (14; 16).

2.8 Analýza konkurence

Požadavkům zadavatele by se jistě dalo částečně vyhovět funkcími klasického účetního systému. Takové řešení by ale nepodporovalo komunikaci s nájemníky, která má největší časovou režii. Proto jsem se zaměřil na informační systémy a aplikace specializované na správu nemovitostí.

2.8.1 DES (Domovní Evidenční Systém)

Na trhu je tento informační systém po dobu devatenácti let. V současné době ho používá přes 500 uživatelů, hlavně městské úřady a bytová družstva. Systém je modulově stavěný, včetně speciálního modulu pro městské úřady, úřady městských částí a magistráty. DES udržuje aktuální stav legislativy a reaguje s předstihem na změny budoucí a změny týkající se způsobu zpracování dat. Program pracuje pouze lokálně, bez webové nadstavby a neumožňuje komunikaci s nájemníky. Existuje možnost provozovat systém na více stanicích, ovšem komunikace mezi nimi probíhá pomocí disket, což může být v dnešní době poněkud problematické. Aktualizace systému fungují přes paušálně placenou službu (17).

Tabulka 3: Základní ceník programové řady DES

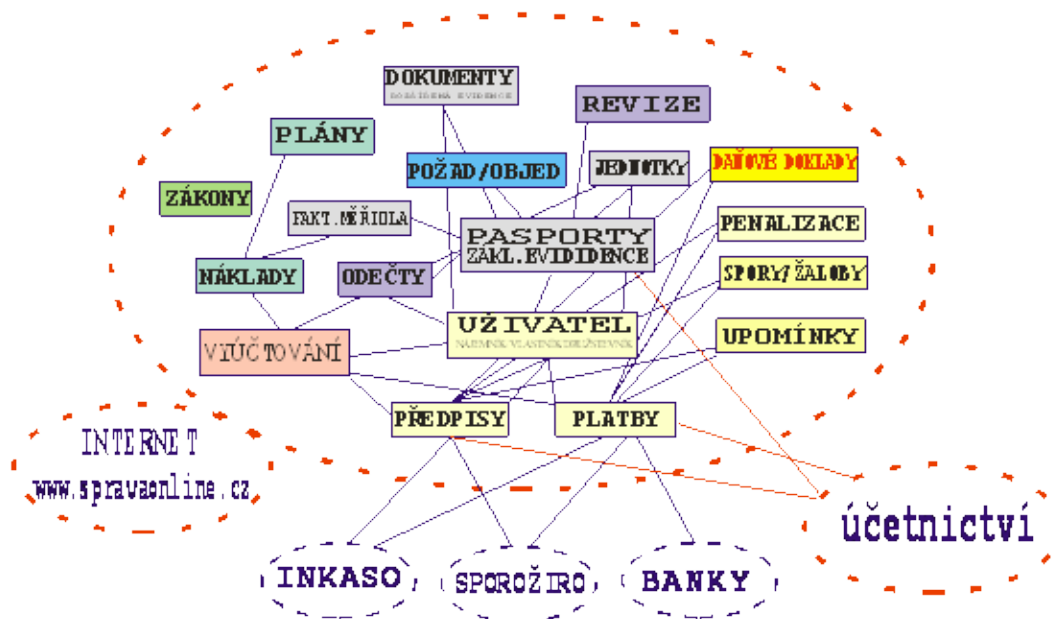
(Zdroj: 17)

Podsystemy programu DES	do 500 nájemních jednotek	do 1000 nájemních jednotek	za každých dalších tisíc nájemních jednotek plus 10% ze základní ceny k základní ceně
Pasport	6000 Kč	9000 Kč	900 Kč
Nájemné	13000 Kč	19000 Kč	1900 Kč
Vyúčtování	13000 Kč	19000 Kč	1900 Kč
Vyúčtování	19000 Kč	29000 Kč	2900 Kč
Účetnictví	9000 Kč	12000 Kč	1200 Kč
Ekonomika domů	3000 Kč	7000 Kč	700 Kč

2.8.2 Windomy

Windomy jsou komplexní software pro správu domů vytvořený firmou OK-SOFT s.r.o., instalovaný z CD, je tedy lokální a neumožňuje komunikaci s nájemníky. Skládá se z modulů, díky čemuž je možné objednat si pouze potřebné moduly a ušetřit. Windomy je možné napojit na účetní systémy. Podle webové prezentace firmy ho používá více než 350 firem spravujících přes 250000 bytových i nebytových prostor v ČR (18).

Nad Windomy existuje také webová aplikace Domy-online.cz, umožňující komunikaci mezi správcem systému, majitelem objektu a uživateli prostoru. Správce má přístup ke všem datům a může přidávat k jednotlivým objektům libovolné dokumenty (18).



Obrázek 6: Podrobný popis aplikace WINDOMY

(Zdroj: www.oksoft.cz/App_Images/popis.gif)

Základní vlastnosti systému zahrnují:

- Technická evidence domů – plochy, vybavení, ...
- Evidence dokumentů – může obsahovat výkresy, fotografie, kopie smluv, dokumenty Word a Excel a další
- Evidence uživatelů – nájemníci a spolumajitelé
- Evidence sporů a jejich průběhu – žaloby, napomenutí
- Umožňuje vytvářet výpisy dle vlastního návrhu
- Správa měřidel a odečtů (18)

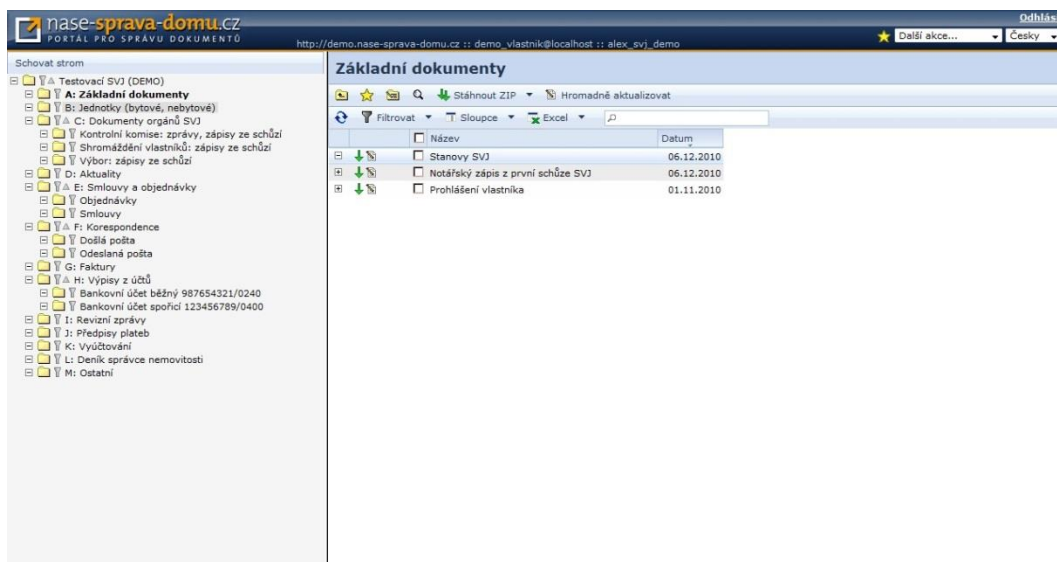
Cena systému závisí na počtu spravovaných bytových a nebytových prostor, počtu instalací a používaných modulech.

Poč. bytů	Zakl.ev	Před/platby	Náklady	Odečty	Požad/objed.	Revize	Vyúčtov.	Vše	sír.st.
0-25	895	1570	314	236	140	84	1180	3977	15%
26-50	1112	1950	390	293	380	228	1290	5078	15%
51-100	2046	3590	718	539	390	234	2240	8781	15%
101-200	2612	4582	916	687	690	414	3150	11746	15%
201-300	3585	6290	1258	944	980	588	4100	15970	15%
301-500	4412	7740	1548	1161	1270	762	5780	20406	15%
501-800	5005	8780	1756	1317	1340	804	7623	23962	15%
801-1000	6145	10780	2156	1617	1740	1044	8082	28407	15%
1001-1500	6669	11700	2340	1755	1990	1194	9882	31977	15%
1501-2000	8436	14800	2960	2220	2350	1410	10782	31146	15%
2001-2500	9103	15970	3194	2396	2740	1644	11691	42064	15%
2501-3000	10249	17980	3596	2697	2847	1708	12140	46095	15%
3001-3500	11252	19740	3948	2961	3040	1824	14382	51432	15%
3501-4000	12962	22740	4548	3411	3170	1902	15282	57613	15%
4001-4500	13659	23964	4793	3595	3540	2124	16182	61071	15%
4501-5000	14478	25400	5080	3810	3804	2282	17082	64743	15%
5000-6000	17043	29900	5980	4485	4120	2472	20800	76320	15%
6000-7000	18639	32700	6540	4905	4390	2634	22900	83437	15%
7000-8000	19893	34900	6980	5235	4580	2748	25700	90032	15%
7000-8500	20777	36450	7290	5468	4890	2934	28750	95902	15%

Obrázek 7: Ceník programového vybavení Windomy (bez DPH)
(Zdroj: <http://www.oksoft.cz/Download/cenik.pdf>)

2.8.3 nase-sprava-domu.cz

Webový informační systém Naše správa domu lze nalézt na adrese nase-sprava-domu.cz. Služba umožňuje bytovým družstvům a vlastníkům organizaci dokumentů a možnost jejich sdílení s nájemníky. První rok provozu je zdarma poté každý další rok stojí 2750 Kč bez DPH (19).



Obrázek 8: Uživatelské rozhraní nase-sprava-domu.cz
(Zdroj: Vlastní)

2.9 SWOT analýza

Poznatky z analýzy konkurence a navrhovaných vlastností jsem shrnul pomocí SWOT analýzy.

Tabulka 4: SWOT analýza
(Zdroj: Vlastní)

Vnitřní prostředí	Silné stránky (Strengths)	Slabé stránky (Weaknesses)
	<ul style="list-style-type: none">• On-line• Moderní design• Jednoduchost• Komunikace s realitními kanceláři• Komunikace s uživateli• Správa dokumentů	<ul style="list-style-type: none">• Snaha proniknout na zavedené trhy• Neobsahuje účetní část• Lze se obejít i bez něj
Vnější prostředí	Příležitosti (Opportunities)	Hrozby (Threats)
	<ul style="list-style-type: none">• Integrace nových technologií• Úprava pro majitele více domů	<ul style="list-style-type: none">• Vstup nové konkurence na trh• Zpoždění příchodu na trh s novým řešením• Modernizace substitutů• Bezpečnostní díry v použitých technologiích

3 Vlastní návrhy řešení a jejich přínos

V této části se věnuji určení ceny systému, zdůvodnění použitých technologií, popsání zajímavých částí kódu a určení praktického použití navrženého informačního systému.

3.1 Cena systému

Cenu informačního systému jsem určil z odhadované časové náročnosti projektu.

I když se investice může zdát poměrně vysoká, pokud informační systém ušetří majiteli domu hodinu týdně, která odpovídá snížení časové náročnosti správy domu o 10 až 14 procent. Časová návratnost investice do informačního systému vychází na 40 týdnů. Dá se však předpokládat vyšší snížení časové náročnosti správy domu a tím i zvýšit návratnost investice.

Cenu systému by bylo možné snížit finančním zapojením dodavatele do projektu, protože dle mého názoru by informační systém mohl zajímat více majitelů bytových domů.

Tabulka 5: Rozpočet
(Zdroj: Vlastní)

Analýza požadavků	5000 Kč
Návrh grafiky webu	5000 Kč
Implementace	30000 Kč
Hosting	299 Kč/měsíčně
Pronájem domény	180-230 Kč/ročně

3.2 Zvolené technologie

Technologie jsem vybíral s ohledem na běžně používané technologie na českých hostinzích. Tím se se snažím vyhnout problémům při výběru hostingu. Dalším kritériem byly moje zkušenosti s těmito technologiemi.

Jako databázový systém jsem zvolil MySQL CE. Se systémem mám již několikaleté zkušenosti. Hlavním důvodem zvolení tohoto produktu je jeho

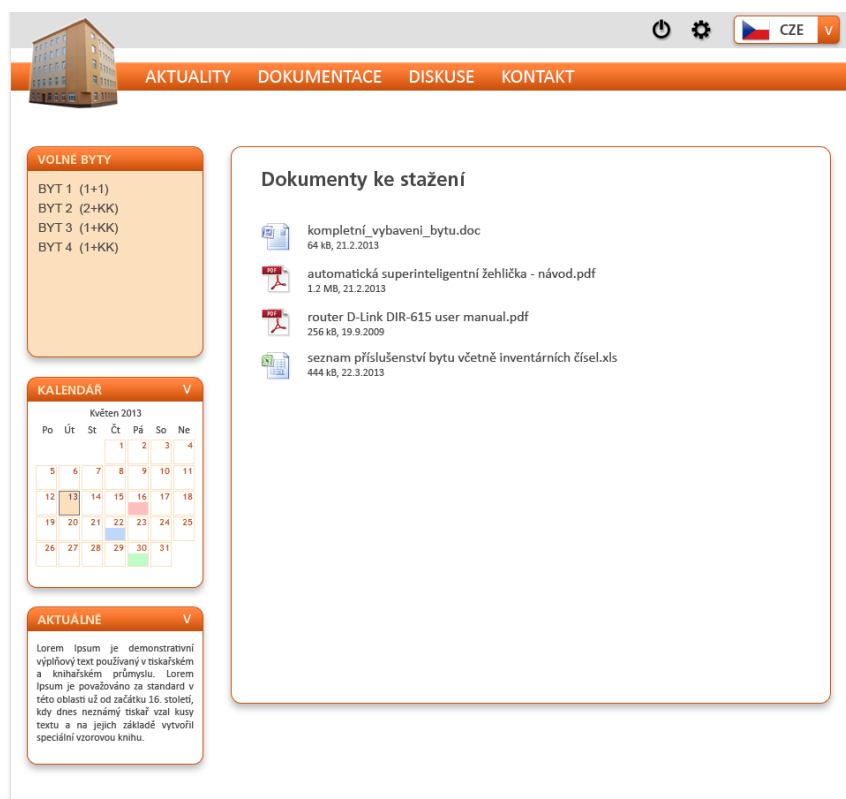
hromadné použití na českých hostingových serverech, které mají ostatní databáze v nabídce většinou jako alternativní produkty.

Použití frameworku není vyloženě nutností, ale ze svých zkušeností mohu potvrdit, že užití frameworku zrychluje vývoj. Rozhodl jsem se využít framework Nette, se kterým mám již zkušenosti.

Z preprocesorů CSS jsem zvolil Less s použitím PHP implementace Less PHP, díky které není nutné instalovat žádné další technologie na server.

3.3 Grafický návrh

Díky CSS je možné vytvořit různé grafické návrhy. Tento návrh je jeden z možných a může být upraven podle požadavků zadavatele systému.

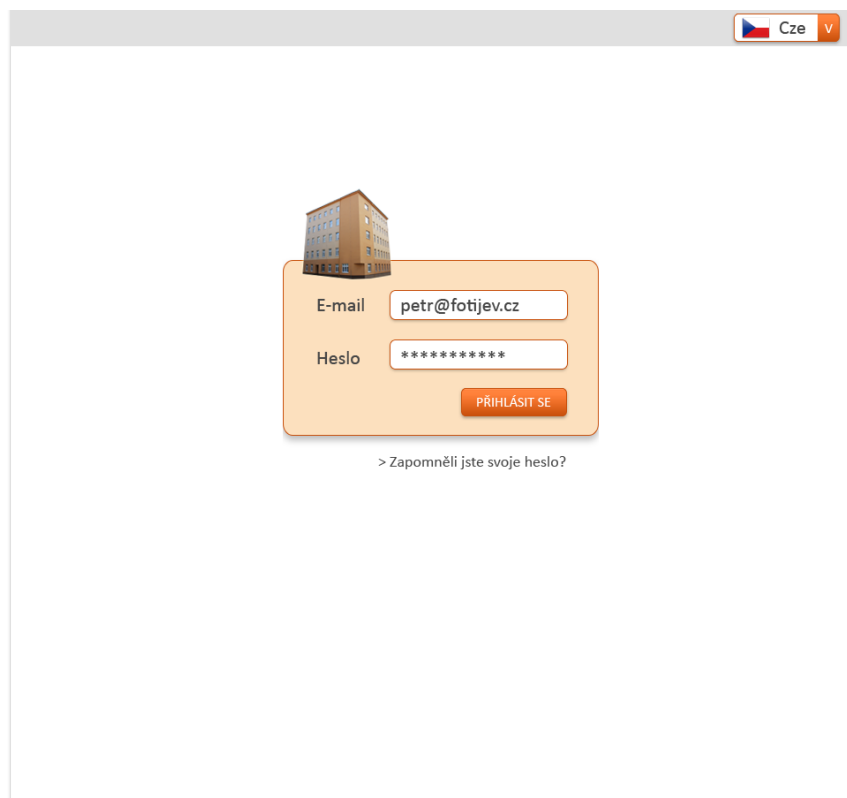


Obrázek 9: Návrh hlavní obrazovky
(Zdroj: Vlastní)

Horní část obsahuje tlačítko pro odhlášení, volbu jazyka a tlačítko nastavení, které umožňuje změnu hesla a údajů. Níže se nachází hlavní menu, které je statické. V levém sloupci se nachází interaktivní menu, jehož obsah se mění v závislosti na

roli uživatele a oprávnění. Realitní kancelář zde vidí seznam volných bytů, kdežto nájemníkovi se zobrazí jeho menu. Pod interaktivním menu je kalendář zobrazující úklid a opravy, které jsou barevně rozlišeny. V kalendáři mohou být zobrazeny v budoucnu i další události. Na kalendář navazuje rychlý přehled aktualit. Počet zobrazených aktualit může záviset na nastavení. Aktuality i kalendář je možné částečně skrýt pomocí ikonky v pravém horním rohu. Největší plochu zabírá obsahová část, kde se budou zobrazovat náplň jednotlivých presenterů.

Přihlašovací obrazovka je nenápadná a neprozrazuje nic o obsahu, kam se uživatel přihlašuje. V pravém horním rohu je pouze volba jazykového nastavení. Použitý obrázek je z fotografie pořízené majitelem, kterou jsem dostal s podklady pro vytvoření grafického návrhu.



Obrázek 10: Návrh logovací obrazovky
(Zdroj: Vlastní)

3.4 Ukázky implementace

V této kapitole bych rád ukázal zajímavé části z implementace informačního systému.

3.4.1 Nastavení routování

Obousměrné routování, použité v Nette znamená, že je možné z URL poznat presenter a jeho akci ale také naopak z akce a presenteru vytvořit URL. Routovací pravidla jsou procházena postupně v pořadí, v jakém jsou přidána internímu routeru.

```
$container->router[] = new Route('index.php',  
    'Front:Homepage:default', Route::ONE_WAY);  
  
$container->router[] = new  
Route('admin/<presenter>/<action>[/<id>]', array(  
    'module' => 'Admin',  
    'presenter' => 'Default',  
    'action' => 'default',  
    'id' => NULL));  
  
$container->router[] = new Route('<presenter>/<action>[/<id>]',  
array(  
    'module' => 'Front',  
    'presenter' => 'Homepage',  
    'action' => 'default',  
    'id' => NULL));  
  
$container->router[] = new Route('<presenter>/<action>[/<id>]',  
    'Front:Homepage:default');
```

Na prvním úseku kódu je možné vidět přesměrování na úvodní stránku z index.php, který je standardně načítán serverem jako výchozí stránka. Druhý úsek obsahuje nastavení směrování modulu administrace. Pokud za adresu webu přidáme /admin budeme přesměrováni do administračního modulu. Všechny presentery v modulu administrace mají v adrese tuto předponu. Následuje nastavení modulu „předku“, u kterého se v adrese nespecifikuje název modulu. Nikdo tedy neví, že nějaké moduly vůbec existují. Kód v poslední části způsobí přesměrování na domovskou stránku v případě zadání neexistujícího presenteru nebo akce.

Pokud adrese nebude vyhovovat žádné pravidlo nebo bude odkazovat na neexistující presentery či akce, zobrazí se stránka jako 404 Not Found.

3.4.2 Nastavení zabezpečení

Autentizace uživatele probíhá po zadání pomocí uživatelského jména a hesla. Heslo je drženo v databázi v zahashované podobě hashovacím algoritmem md5 s použitím soli. Z databáze podle uživatelského jména jsou získány záznamy o uživateli včetně jeho role a hesla. Zadané heslo se zahashování porovná s heslem z databáze. Po autentizaci jsou informace o uživateli uloženy.

```
public function authenticate(array $credentials) {
    list($username, $password) = $credentials;
    $row = $this->database->table('users')->where('username',
    $username)->fetch();
    if (!$row) {
        throw new Security\AuthenticationException("User '$username'
        not found.", self::IDENTITY_NOT_FOUND);
    }

    \Nette\Diagnostics\Debugger::fireLog(self::calculateHash(
    $password, $row->password));

    if ($row->password !== self::calculateHash($password,
    $row->password)) {
        throw new Security\AuthenticationException(
        "Invalid password.", self::INVALID_CREDENTIAL);
    }

    $role = $this->database->table('roles')->get($row->role);

    unset($row->password);
    return new Security\Identity($row->id, $role->name,
    $row->toArray());
}
```

Všechny presentery, jejichž obsah má být chráněn heslem, kontrolují uložené informace o uživateli. Pokud žádné informace uloženy nejsou, uživatel není přihlášen a je přesměrován na přihlašovací stránku. Obdobně fungují omezení rolmi uživatelů, z uložených informací se získá role uživatele a porovná se s povolenými rolmi. Pokud role uživatele není v povolených rolích, je uživatel přesměrován na úvodní stránku.

3.4.3 Lokalizace

K lokalizaci je použito rozšíření frameworku Nette GettextTranslator. Rozšíření funguje jako služba, která se přiřadí komponentě. Poté přistupuje ke všem textům směřujícím na výstup jako k heslům, které hledá ve slovníku. Překlad v šablonách

probíhá užitím direktivy {_"week",5}, kde week je heslo a 5 je čítač množného čísla, podle kterého se určí správný tvar odpovídající dané mohutnosti.

```
if (!isset($this->lang)) {  
    $this->context->parameters["lang"] = 'cs';  
    $this->lang = $this->context->parameters["lang"];  
}  
$this->context->translator->setLang($this->lang);
```

Na ukázce je zobrazeno nastavení jazykové mutace. Takto je přidáno jazykové nastavení do adresního řádku webu. Jazyková mutace může být kdykoliv změněna pouze nastavením jiného jazyka překladače a obnovením stránky.

```
"Plural-Forms: nplurals=3; plural=((n==1) ? 0 : (n>=2 && n<=4 ? 1  
: 2));\n"
```

```
msgid "week"  
msgid_plural "week"  
msgstr[0] "týden"  
msgstr[1] "týdny"  
msgstr[2] "týdnů"
```

Začátek každého slovníku obsahuje nastavení a vlastnosti, je nutné nastavit mezi jinými vlastnostmi pravidla množných čísel jazyku překladače. V ukázce je možné vidět pravidla českého jazyka. Následují zápisy a překlady jednotlivých hesel. Definování množných čísel není povinné. Jazykové slovníky mají podobu dvou souborů, které jsou vzájemně mezi sebou převoditelné:

- První má koncovku „po“, která zastupuje zkrácené person object. V tomto souboru se nachází slovník v textové podobě čitelné lidmi.
- Druhá koncovka „mo“ což je zkratka machine object je přeložený první soubor v binární podobě. Tento soubor používá Gettext pro překlad, protože práce s ním je rychlejší

3.4.4 Ukládání obrázků galerií

Po odeslání souboru je obrázek uložen do specifikované složky a je přejmenován podle aktuálního data a času, tím se zabrání duplikaci názvů. Obrázek je poté otevřen nástrojem pro zpracování obrázků, který ho zmenší tak aby se vešel do čtverce o straně 1024 pixelů, bez změny poměru stran obrázku. Výsledný obrázek je vyostřen a uložen. Z tohoto obrázku je následně vytvořen čtvercový náhled

použitý v galerii. Ten vytvořím zmenšením menší strany na rozměr 192 pixelů, vycentrováním čtverce o straně 192 pixelů a oříznutí přesahů. Následně jsou uloženy informace o obrázcích do databáze.

```
public function formSubmitted(Form $form) {
    $file = $form['file']->getValue();
    if ($file->isOK()) {
        $name = date('YmdHis');
        $file->move($this->path . $name.'.jpg');

        $image = Image::fromFile($this->path . $name . '.jpg');
        $image->resize($this->max_dimension, $this->max_dimension);
        $image->sharpen();
        $image->save($this->path . $name . '.jpg');
        $image->resize($this->thumbnailDimension, $this->thumbnailDimension, Image::EXACT);
        $image->save($this->path . $name . '-thn.jpg');

        $images = $this->getPresenter()->getContext()
            ->createFlatsImages();
        $images->insert(array('file' => $name,
                            'width' => $image->width,
                            'height' => $image->height,
                            'fk_flats' => $this->record));
    } else {
        $this->getPresenter()
            ->flashMessage('Failed_file_upload');
    }
    $this->redirect('this');
}
```

3.5 Praktické využití

Dá se předpokládat postupné zvykání nájemníků na užívání informačního systému. Předpokládám snížení počtu opakujících se dotazů a problémů, které majitel domu řeší nejčastěji. S postupným zvyšováním užívání systému se bude zvyšovat i jeho účinnost.

Sebelépe navržený a naprogramovaný informační systém bude fungovat pouze tak dobře, jak se mu bude provozovatel věnovat. Proto je nutné, aby data v systému byla aktuální a pravdivá. Toto však záleží na provozovateli informačního systému.

Použití systému je možné zahrnout i v nájemní smlouvě tak, že informace zobrazené v informačním systému budou po dané době brány jako přečtené a nájemník je s nimi obeznámen. Toto využití navíc podporuje log přístupů do systému. Tímto krokem je možné donutit nájemníky využívat informační systém, a tím zvýšit jeho účinnost.

4 Závěr

V teoretických východiscích práce jsem objasnil význam pojmu správa domu, objasnil základní pojmy problematiky a obecně popsal informační systém. Věnoval jsem se popisu přístupů k projektům softwarového vývoje. Prozkoumal jsem nové trendy webových technologií HTML a CSS a organizaci W3C zabývající se standardizováním těchto technologií. Závěrem jsem popsal architekturu webových prezentací MVC.

Na základě vyhodnocených požadavků majitele domu jsem sestavil specifikaci funkcí, které by si majitel přál a přidal jsem funkčnosti, které z nich vyplývaly. Snažil jsem se přidat funkčnost, kterou majitel nespecifikoval, ale mohla by být užitečná. Podle vytvořené specifikace jsem vytvořil podklady pro implementaci a seznámení zadavatele s podobou budoucího systému. Dále jsem vytvořil vývojový plán projektu.

Zkoumáním konkurence jsem došel k závěru, že na trhu není informační systém, který by odpovídal požadavkům zadavatele. Nejblíže k požadavkům zadavatele je informační systém Windomy s nadstavbou Domy-online.cz, i když je systém dle mého názoru zbytečně robustní pro majitele jednoho domu. Informační systém DES je určený hlavně pro města, komunikace s nájemníkem neexistuje a je celkově zastaralý. Webový portál Naše správa domu by požadavkům komunikace s nájemníky rámcově vyhovoval, ale celkově je tento systém pouze ořezaným řešením.

Použité technologie jsem vybíral s ohledem na standardní vybavení českých hostingů a své zkušenosti. Bez frameworku Nette a preprocesoru CSS Less PHP by se vývoj mohl obejít. Tyto nástroje však zvyšují efektivnost a znovupoužití kódu a dnes jsou běžně používány. Jako databázový stroj jsem zvolil MySQL CE.

Část návrhu, jehož funkčnost jsem se rozhodl implementovat, se týká realitní kanceláře a její administrátorské části včetně technologie zabezpečující jazykové mutace. Důležité nebo zajímavé implementace jsem popsal.

Seznam použitých zdrojů

1. TARABA, Milan a Lenka VESELÁ. *Rádce nájemníka bytu*. 7., zcela přeprac. vyd. Praha: Grada, 2012, 250 s. ISBN 978-802-4744-056.
2. BÉBR, Richard a Petr DOUCEK. *Informační systémy pro podporu manažerské práce*. Praha: Profesional publishing, 2005. 223 s. ISBN 80-86419-79-7.
3. DOUCEK, Petr. *Řízení projektů informačních systémů*. Praha: Profesional publishing, 2004. 163 s. ISBN 80-86419-71-1.
4. BRUCKNER, Tomáš, et al. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. ISBN 978-80-247-4153-6.
5. WORLD WIDE WEB CONSORCIUM. *World Wide Web Consortium* [online]. 2013. vyd. [cit. 2013-05-10]. Dostupné z: <http://www.w3.org/>
6. Přehled standardů W3C. *Interval.cz* [online]. 27. 05. 2002. 2002 [cit. 2013-05-15]. Dostupné z: <http://interval.cz/clanky/prehled-standardu-w3c/>
7. VAUGHAN-NICHOLS, S. J. Will HTML 5 Restandardize Web?. *Computer* [online]. 2010, vol. 43, no. 4, s. 13-15 [cit. 2013-05-15]. ISSN 0018-9162. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5445161>
8. CSS3 Previews. *CSS3.info: All you ever needed to know about CSS3* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://www.css3.info/preview/>
9. NETTE FOUNDATION. *Nette Framework* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://nette.org/cs/>
10. Edice Microsoft SQL Server Express. MICROSOFT. *Microsoft SQL Server* [online]. 2013 [cit. 2013-05-05]. Dostupné z: <http://www.microsoft.com/sqlserver/cs/cz/editions/express.aspx>
11. MySQL Community Edition. SUN MICROSYSTEMS. *MySQL* [online]. 2013 [cit. 2013-05-14]. Dostupné z: <http://www.mysql.com/products/community/>

12. About. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP.
PostgreSQL [online]. 2013 [cit. 2013-05-14]. Dostupné z:
<http://www.postgresql.org/about/>
13. About Zend Framework. ZEND TECHNOLOGIES. *Zend Framework*
[online]. 2013 [cit. 2013-05-09]. Dostupné z:
<http://framework.zend.com/about/>
14. CSS preprocesory: méně psaní, vyšší efektivita. In: JAVOREK, Jan. *Zdroják*
[online]. 29.4.2011. 2011 [cit. 2013-05-13]. Dostupné z:
<http://www.zdrojak.cz/clanky/css-preprocesory-mene-psani-vyssi-efektivita/>
15. *Sass* [online]. 2013 [cit. 2013-05-13]. Dostupné z: <http://sass-lang.com/>
16. *LESS* [online]. 2013 [cit. 2013-05-13]. Dostupné z: <http://lesscss.org/>
17. Domovní Evidenční Systém. TOM COMPUTER. *Domovní Evidenční Systém*
[online]. 2011 [cit. 2013-05-04]. Dostupné z: <http://www.destom.cz/>
18. Windomy. O.K. SOFT. *O.K. Soft* [online]. 2012 [cit. 2013-05-04]. Dostupné
z: <http://www.oksoft.cz/>
19. Naše správa domu. EXCON. *Naše správa domu* [online]. 2013
[cit. 2013-05-04]. Dostupné z: <http://nase-sprava-domu.cz/>

Seznam obrázků

Obrázek 1: Vodopádový model životního cyklu projektu	20
Obrázek 2: Spirálový model	21
Obrázek 3: Interakce uživatele s architekturou MVC	26
Obrázek 4: Use Case diagram.....	30
Obrázek 5: ER diagram	32
Obrázek 6: Podrobný popis aplikace WINDOMY	38
Obrázek 7: Ceník programového vybavení Windomy (bez DPH)	39
Obrázek 8: Uživatelské rozhraní nase-sprava-domu.cz	40
Obrázek 9: Návrh hlavní obrazovky	43
Obrázek 10: Návrh logovací obrazovky	44

Seznam tabulek

Tabulka 1: Uživatelé systému.....	16
Tabulka 2: Fáze vývoje.....	33
Tabulka 3: Základní ceník programové řady DES	37
Tabulka 4: SWOT analýza	41
Tabulka 5: Rozpočet.....	42

Seznam příloh

Příloha A – CD se zdrojovými kódy

Příloha A – CD se zdrojovými kódy

Obsah:

- Složka src – zdrojové soubory implementace
- Soubor database.sql – SQL skript na vytvoření databáze